Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L4: Entry 2 of 7 File: USPT Jan 21, 2003

DOCUMENT-IDENTIFIER: US 6510158 B1
TITLE: Method and apparatus combining a plurality of virtual circuits into a combined virtual circuit

Abstract Text (1):
A method and apparatus includes processing for combining a plurality of virtual circuits into a combined virtual circuit, where such processing begins by buffering cells of each virtual circuit into a corresponding buffer. The processing then continues by obtaining priority information for each virtual circuit and obtaining logical buffer de-queuing information. The priority information, for example, may equate to priorities established via the varying levels of ATM services. The logical buffer de-queuing information corresponds to an access sequence for a plurality of logical ring buffers that are comprised of the buffers, or buffer identifiers. The processing then continues by generating the combined virtual circuit based on the logical buffer de-queuing information and the priority information.

Application Filing Date (1):
19990430

Detailed Description Text (2):
Generally, the present invention provides a method and apparatus that includes processing for combining a plurality of virtual circuits into a combined virtual circuit. Such processing begins by buffering cells of each virtual circuit into a corresponding buffer. The processing then continues by obtaining priority information for each virtual circuit and obtaining logical buffer de-queuing information. The priority information, for example, may equate to priorities established via the varying levels of ATM services. The logical buffer de-queuing information corresponds to an access sequence for a plurality of logical ring buffers that are comprised of the buffers, or buffer identifiers. The processing then continues by generating the combined virtual circuit based on the logical buffer de-queuing information and the priority information. With such a method and apparatus, virtual circuits within a virtual circuit merge connection or a virtual path aggregation connection can be generated with varying degrees of priority, which may be established in accordance with the ATM services associated with the each of the virtual circuits.

Detailed Description Text (7):
The endpoint switch 50 also includes a plurality of logical buffers 62 wherein each of the logical buffers stores an access sequence based on varying priorities of the currently received virtual circuits 56. As shown, the endpoint switch 50 currently has five logical buffers corresponding to virtual circuits 56. The access sequence is stored in each of the logical buffers 62. As such, a first access sequence is ABCDEF, a second one is ACBDEF, a third ADBCEF, a fourth AEBCDF, and a fifth AFBCDE. For this illustration, the access sequence stored in the third logical buffer of ADBCEF is utilized. Accordingly, based on the cells stored in buffers A-F and the access sequence of the third logical buffer, the endpoint switch 50 generates a combined virtual circuit 58. The number of cells accessed and placed in the combined virtual circuit is dependent on the service interval 60. In this

illustration, the service interval is six cells. As one of average skill in the art will appreciate, the service interval may range from a single cell to hundreds of cells.

Detailed Description Text (8):
To produce the combined virtual circuit 58, the endpoint switch 50 accesses the third logical buffer 62 and determines that the highest priority virtual circuit is A. Based on this information, the endpoint switch 50 accesses buffer A to retrieve as many cells stored in buffer A up to the service interval length. In this illustration, buffer A is storing two cells. As such, the first two cells of the combined virtual circuit for the current service interval are filled with cells of virtual circuit A. The endpoint switch then accesses the logical buffer to determine that the next highest priority virtual circuit is D. The endpoint switch 50 access buffer D to retrieve the two cells for virtual circuit D and places those in the next two cells of the service interval. Having done this, the endpoint switch 50 then accesses the logical buffer 62 to determine that the next highest priority virtual circuit is B. The endpoint switch 50 accesses buffer B to retrieve cells stored in buffer B. Endpoint switch 50 determines that only two cells can be retrieved to complete the service interval 60. As such, endpoint switch only retrieves two cells from buffer B. On the next service interval, endpoint switch 50 again accesses the logical buffer 62 and determines that the highest priority virtual circuit is A. If no new cells for virtual circuit A have been received during the previous service interval, the endpoint switch traverses the logical buffer 62 until it finds a buffer A-F that contains the next highest priority level virtual circuit and has cells to be transported.

Detailed Description Text (14):
As one of average skill in the art will appreciate, the number of buffers used by a switch to produce a combined virtual circuit is dependent on the number of virtual circuits to be combined. Further, the plurality of logical buffers is also dependent upon the number of virtual circuits to be combined and can be derived by the intervening switch based on priorities prescribed to each virtual circuit, which may be in accordance with ATM priorities such as cost, bandwidth, latency, etc. As one of average skill in the art will further appreciate, each of the buffers and logical buffers may be established as ring buffers that contain a head pointer and tail pointer such that from service interval to service interval, the intervening switch can readily determine which cells are to be provided in the current combined virtual circuit. In addition, the access sequence contained in the logical buffers may be changed from service interval to service interval, may remain fixed for a certain number of service intervals, or may remain fixed until changed by a network administrator.

Detailed Description Text (16):
FIG. 5 illustrates a logic diagram of a method for combining a plurality of virtual circuits into a combined virtual circuit. The process begins at step 110 where cells for each virtual circuit are buffered in a corresponding buffer. The process then proceeds to step 112 where priority information is obtained for each virtual circuit. The priority ~~information for each virtual circuit includes a logical ring~~ buffer identifier and a weighting factor relative to a logical ring buffer of a plurality of logical ring ~~buffers. This was illustrated with reference to~~ FIGS. 2 and 3 where the plurality of logical buffers included an access sequence wherein the buffer was identified by the logical ring buffer identifier (e.g., A, B, C, etc.).

Detailed Description Text (17):
The process then proceeds to step 114 where logical buffer de-queuing information is obtained for generating a combined virtual circuit. The logic buffer de-queuing information includes an access sequence for each of the plurality of ring buffers. Again, this was illustrated with reference to FIGS. 2 and 3. The process then continues to step 116 where the combined virtual circuit is generated based on the

logical buffer de-queuing information and the priority information. Note that the combined virtual circuit may be a virtual path aggregation or a virtual circuit merge. When the combined virtual circuit is based on a virtual circuit merge, the combined virtual circuit is generated based on an end-of-message detection for each virtual circuit and the cells for each virtual circuit are transported per service interval. As one of average skill in the art will appreciate, an end-of-message, which may be indicating the end of a higher level protocol packet, is contained in the last cell sent before starting service of another virtual circuit. In addition, if a virtual circuit merge is being performed, then cells of a component virtual circuit are only considered for transmission after a cell containing the end-of-message indication has been queued. One of average skill in the art will further appreciate that the use of weighting factors between virtual circuits may be used on the same circular buffer.

Detailed Description Text (18):
The logical buffer de-queuing information used to produce the combined virtual circuit may be established in a round-robin manner based on weighting factors for each of the plurality of virtual circuits. The round robin ordering constitutes the plurality of logical ring buffers. To facilitate the accessing of ring buffers, each ring buffer includes a head and/or tail pointer. The de-queuing information further indicates that a cell from a next priority level logic ring buffer will be accessed when the current priority logical ring buffer is empty. This was illustrated with reference to FIGS. 2 and 3.

Detailed Description Text (19):
FIG. 6 illustrates a logic diagram of an alternate method for combining a plurality of virtual circuits into a combined virtual circuit. The process begins at step 120 where cells from each virtual circuit are buffered in a corresponding buffer. The process then proceeds to step 122 where de-queuing of the cells is based on priority of each virtual circuit. Such de-queuing may be done in an allocated manner such that the de-queuing sequence varies to provide adequate priority accessing. The process then proceeds to step 124 where the combined virtual circuit is produced as the cells are de-queued. Note that the combine virtual circuit may be produced based on an end-of-message detection for a virtual circuit and such cells for the virtual circuit are transported per service interval. Alternatively, the de-queuing of the cells from the plurality of buffers may be done in a round-robin manner based on a weighting factor for each virtual circuit where the de-queuing order constitutes a plurality of logical ring buffers. When de-queued in this manner, a cell from a next prior level logical ring buffer is de-queued when the current priority logical ring buffer is empty. With respect to a virtual circuit merge, the de-queuing may be based on a combination of EOM messages and weighted round-robin, where the weights might correspond to the number of fall packets a connection gets to send (assuming it has at least this many packets buffered) before going to the next connection in the ring buffer, or the weight may be a minimum number of cells guarantee whereby the connection will send at least this many cells up to the next EOM (or all of the full packets that are currently buffered for this connection, whichever is fewer cells) before the ring buffer switches to the next connection. These are just two of the many possibilities.

CLAIMS:

1. A method for combining a plurality of virtual circuits into a combined virtual circuit, the method comprises the steps of: a) buffering cells of each of the plurality of virtual circuits into a corresponding one of a plurality of buffers; b) obtaining priority information for each of the plurality of virtual circuits; c) obtaining logical buffer dequeuing information for the combined virtual circuit; and d) generating the combined virtual circuit based on the logical buffer dequeuing information and the priority information of each of the plurality of virtual circuits, wherein the combined virtual circuit is at least one of a virtual path aggregation and a virtual circuit merge wherein step (d) further comprises

generating the virtual circuit merge based on an end of message detection for a virtual circuit of the plurality of virtual circuits and cells transported per service interval for the virtual circuit.

2. A method for combining a plurality of virtual circuits into a combined virtual circuit, the method comprises the steps of: a) buffering cells of each of the plurality of virtual circuits into a corresponding one of a plurality of buffers; b) obtaining priority information for each of the plurality of virtual circuits; c) obtaining logical buffer dequeuing information for the combined virtual circuit; and d) generating the combined virtual circuit based on the logical buffer dequeuing information and the priority information of each of the plurality of virtual circuits, wherein the priority information for each of the plurality of virtual circuits comprises a logical ring buffer identifier and a weighting factor relative to a logical ring buffer of a plurality of logical ring buffers identified by the logical ring buffer identifier.

3. The method of claim 2, wherein the logical buffer dequeuing information comprises an accessing sequence for the plurality of logical ring buffers.

4. The method of claim 3, wherein step (c) further comprises dequeuing cells from the plurality of buffers in a round robin manner based on the weighting factors of the plurality of virtual circuits, wherein ordering of the plurality of buffers constitutes the plurality of logical ring buffers.

5. The method of claim 4, further comprises maintaining head and/or tail pointers of each of the plurality of logical ring buffers.

6. The method of claim 4, further comprises dequeuing a cell from a next priority level logical ring buffer of the plurality of logical ring buffers when the current priority logical ring buffer of the plurality of logical ring buffers is empty.

7. A method for combining a plurality of virtual circuits into a combined virtual circuit, the method comprises the steps of: a) buffering cells of each of the plurality of virtual circuits into a corresponding one of a plurality of buffers; b) dequeuing of the cells based on priority of each of the plurality of virtual circuits and a virtual circuit accessing sequence; and c) producing the combined virtual circuit as the cells are dequeued, wherein the combined virtual circuit is at least one of a virtual path aggregation and a virtual circuit merge, wherein step (c) further comprises, producing the virtual circuit merge based on an end of message detection for a virtual circuit of the plurality of virtual circuits and cells transported per service interval for the virtual circuit.

8. A method for combining a plurality of virtual circuits into a combined virtual circuit, the method comprises the steps of: a) buffering cells of each of the plurality of virtual circuits into a corresponding one of a plurality of buffers; b) dequeuing of the cells based on priority of each of the plurality of virtual circuits and a virtual circuit accessing sequence; and c) producing the combined virtual circuit as the cells are dequeued, wherein the priority for each of the plurality of virtual circuits comprises a logical ring buffer identifier and a weighting factor relative to a logical ring buffer of a plurality of logical ring buffers identified by the logical ring buffer identifier.

9. The method of claim 8, wherein step (c) further comprises dequeuing the cells from the plurality of buffers in a round robin manner based on the weighting factors of the plurality of virtual circuits, wherein ordering of the plurality of buffers constitutes the plurality of logical ring buffers.

10. The method of claim 9, further comprises dequeuing a cell from a next priority level logical ring buffer of the plurality of logical ring buffers when the current priority logical ring buffer of the plurality of logical ring buffers is empty.

11. A communication switch comprises: processing module; and memory operably coupled to the processing module, wherein the memory-includes operational instructions that cause the processing module to (a) buffer cells of each of the plurality of virtual circuits into a corresponding one of a plurality of buffers; (b) obtain priority information for each of the plurality of virtual circuits; (c) obtain logical buffer dequeuing information for the combined virtual circuit; and (d) generate the combined virtual circuit based on the logical buffer dequeuing information and the priority information of each of the plurality of virtual circuits, wherein the combined virtual circuit is at least one of a virtual path aggregation and a virtual circuit merge, wherein the memory further comprises operational instructions that cause the processing module to generate the virtual circuit merge based on an end of message detection for a virtual circuit of the plurality of virtual circuits and cells transported per service interval for the virtual circuit.

12. The communication switch of claim 11, wherein the priority information for each of the plurality of virtual circuits comprises a logical ring buffer identifier and a weighting factor relative to a logical ring buffer of a plurality of logical ring buffers identified by the logical ring buffer identifier.

13. The communication switch of claim 12, wherein the logical buffer dequeuing information comprises an accessing sequence for the plurality of logical ring buffers.

14. The communication switch of claim 13, wherein the memory further comprises operational instructions that cause the processing module to dequeue cells from the plurality of buffers in a round robin manner based on the weighting factors of the plurality of virtual circuits, wherein ordering of the plurality of buffers constitutes the plurality of logical ring buffers.

15. The communication switch of claim 14, wherein the memory further comprises operational instructions that cause the processing module to dequeue a cell from a next priority level logical ring buffer of the plurality of logical ring buffers when the current priority logical ring buffer of the plurality of logical ring buffers is empty.

16. A communication switch comprises: processing module; and memory operably coupled to the processing module, wherein the memory includes operational instructions that cause the processing module to (a) buffer cells of each of the plurality of virtual circuits into a corresponding one of a plurality of buffers; (b) dequeue the cells based on priority of each of the plurality of virtual circuits and a virtual circuit accessing sequence; and (c) produce the combined virtual circuit as the cells are dequeued, wherein the combined virtual circuit is at least one of a virtual path aggregation and a virtual circuit merge, wherein the memory further comprises operational instructions that cause the processing module to produce the virtual circuit merge based on an end of message detection for a virtual circuit of the plurality of virtual circuits and cells transported per service interval for the virtual circuit.

17. A communication switch comprises: processing module; and memory operably coupled to the processing module, wherein the memory includes operational instructions that cause the processing module to (a) buffer cells of each of the plurality of virtual circuits into a corresponding one of a plurality of buffers; (b) dequeue the cells based on priority of each of the plurality of virtual circuits and a virtual circuit accessing sequence; and (c) produce the combined virtual circuit as the cells are dequeued, wherein the priority for each of the plurality of virtual circuits comprises a logical ring buffer identifier and a weighting factor relative to a logical ring buffer of a plurality of logical ring buffers identified by the logical ring buffer identifier.

18. The communication switch of claim 17, wherein the memory further comprises operational instructions that cause the processing module to dequeue the cells from the plurality of buffers in a round robin manner based on the weighting factors of the plurality of virtual circuits, wherein ordering of the plurality of buffers constitutes the plurality of logical ring buffers.

19. The communication switch of claim 18, wherein the memory further comprises operational instructions that cause the processing module to dequeue a cell from a next priority level logical ring buffer of the plurality of logical ring buffers when the current priority logical ring buffer of the plurality of logical ring buffers is empty.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L4: Entry 5 of 7 File: USPT Apr 7, 1998

DOCUMENT-IDENTIFIER: US 5737514 A
TITLE: Remote checkpoint memory system and protocol for fault-tolerant computer system

Application Filing Date (1):
19951129

Brief Summary Text (19):
In another embodiment of the present invention, the remote checkpoints buffer are provided to allow N+1 redundancy by configuring computers in a logical ring with each computer acting as a backup for one of its adjacent neighbors.

Detailed Description Text (27):
After a failure of one of the computers, the surviving computer 1) may run applications of both computers (itself and the failed computer) with a decrease in throughput for any one application, or 2) may terminate its own applications and only run those of the failed computer, or 3) may run a subset of the combined applications that are of sufficiently high priority.

CLAIMS:

6. A computer system comprising:

at least three computers, each of the computers except one spare computer performing data processing tasks;

a data communications network coupled to each of the computers such that the plurality of computers are connected in a logical ring;

wherein each computer includes a processor having a cache, internal registers and input/output event queue and connected to a main memory and a write buffer coupled to capture data written to the memory by the processor;

means for flushing the cache, internal registers and input/output event queue to the main memory and means for copying data from the write buffer to the main memory of an adjacent computer in the logical ring at periodic checkpoints; and

wherein upon failure of one of the computers performing data processing tasks, the data processing tasks of the computers performing data processing tasks are performed by functional computers of the computers including the spare computer without loss of data.

Previous Doc Next Doc Go to Doc#